Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1. (Currently amended) A computer-implemented method of organizing components to provide access to a first service, comprising:

grouping components together to perform the first service, wherein each component implements an interface for communicating with an assembly manager; and

defining [[an]] a first assembly, the first assembly having a first name and [[an]] a first assembly ~~definition~~ data-structure having metadata information identifying each component in the group of components and any further interfaces implemented by or used by any of the components, the first assembly data-structure determining the components required to provide access to the first service and the relationships between the components, whereby the first assembly ~~definition~~ data-structure is configured to be loaded into the assembly manager, the assembly manager being operable to load and assemble the components at run-time according to the metadata information to provide access to the first service.

2. (Currently amended) The computer-implemented method of claim 1, further comprising associating the first name of the first assembly with a role name associated with the first service.

3. (Currently amended) The computer-implemented method of claim 1, further comprising:

grouping the first assembly with components to perform a second service; and

defining a second assembly, the second assembly having a second name and a second assembly ~~definition~~ data-structure having metadata information identifying the first assembly and each component in the group of components and any further interfaces implemented by or used by the first assembly and any of the components, the second assembly data-structure determining the components required to provide access to the second service and the relationships between the first assembly and the components, whereby the second assembly ~~definition~~ data-structure is configured to be loaded into the assembly manager, the assembly manager being operable to load and assemble the components at run-time according to the metadata information to provide access to the second service.

4. (Currently amended) The computer-implemented method of claim 1, further comprising:

modifying one of the components;

creating a new component to filter information that passes through an interface connected to the modified component; and

modifying the first assembly ~~definition~~ data-structure to specify the new component and an interface connecting the new component to the modified component, whereby the modified component and new component so connected produce filtered information compatible with other components in the first assembly.

5. (Currently amended) The computer-implemented method of claim 4, wherein modifying the component alters the processing of information and renders the modified component and information incompatible with the other components in the first assembly.

6. (Currently amended) The computer-implemented method of claim 1, wherein defining the first assembly comprises identifying client and server relationships between the components and interfaces.

7. (Currently amended) The computer-implemented method of claim 1, wherein the first assembly ~~definition~~ data-structure is represented using Extensible Markup Language (XML).

8. (Currently amended) The computer-implemented method of claim 1, wherein the components and interfaces comply with an object model architecture.

9. (Currently amended) The computer-implemented method of claim 8, wherein the object-model is selected from a set of object-models including Component Object Model (COM), Bravo Interface Binder (BIB), and Common Object Request Broker Architecture (CORBA).

10. (Currently amended) A computer-implemented method of providing access to a service by a component-based application, comprising:

receiving a request from the component-based application that identifies a service;

accessing an assembly ~~definition~~ data-structure associated with the service and having metadata information specifying a number of components used to perform the service and interfaces implemented by and used by the components, the assembly data-structure determining the components required to perform the service and the relationships between the components;

loading each component identified in the assembly data-structure into an area for processing; and

connecting an interface associated with each loaded component to other components according to the meta-data information in the assembly ~~definition~~ data-structure to form an assembly, whereby the application has access to an interface for communicating with the assembly.

11. (Currently amended) The computer-implemented method of claim 10, further comprising:

connecting interfaces identified in the assembly ~~definition~~ data-structure to the loaded components; and

connecting interfaces associated with components in the assembly ~~definition~~ data-structure but not identified in the assembly ~~definition~~ data-structure to the loaded components.

12. (Currently amended) The computer-implemented method of claim 11, further comprising connecting interfaces in the assembly to components in a previously loaded assembly.

13. (Currently amended) The computer-implemented method of claim 10, further comprising:

receiving an indication that the access to the requested service is not longer required;

disconnecting the interface from each component associated with the requested service; and

unloading each disconnected component and the corresponding assembly ~~definition~~ data-structure while the component-based application remains loaded.

14. (Currently amended) A computer-implemented method for gaining access to a service, comprising:

identifying a service for processing data;

calling an assembly manager with a service request corresponding to the service, the assembly manager determining an assembly that is capable of performing the service; and

accessing [[an]] the assembly ~~capable of performing the service~~, the assembly including components and interfaces specified in an assembly ~~definition~~ data-structure and loaded by the assembly manager, the assembly data-structure determining the components required to perform the service and the relationships between the components, the assembly manager being operable to assemble the components at run-time according to information in the assembly data-structure to perform the service.

15. (Currently amended) The computer-implemented method of claim 14, wherein the service request comprises a name associated with the assembly ~~definition~~ data-structure.

16. (Currently amended) The computer-implemented method of claim 14, wherein the service request comprises a role name associated with the service.

17. (Currently amended) A computer program product, tangibly stored on a computer-readable medium, for organizing components to provide access to a first service, the product comprising instructions operable to cause a programmable processor to:

group components together to perform the first service, wherein each component implements an interface for communicating with an assembly manager; and

define [[an]] a first assembly, the first assembly having a first name and [[an]] a first assembly ~~definition~~ data-structure having metadata information identifying each component in the group of components and any further interfaces implemented by or used by any of the components, the first assembly data-structure determining the components required to provide access to the first service and the relationships between the components, whereby the first assembly ~~definition~~ data-structure is configured to be loaded into the assembly manager, the assembly manager being operable to load and assemble the components at run-time according to the metadata information to provide access to the first service.

18. (Currently amended) The product of claim 17, further comprising instructions operable to cause the processor to associate the first name of the first assembly with a role name associated with the first service.

19. (Currently amended) The product of claim 17, further comprising instructions operable to cause the processor to:

group the first assembly with components to perform a second service; and

define a second assembly, the second assembly having a second name and a second assembly ~~definition~~ data-structure having metadata information identifying the first assembly and each component in the group of components and any further interfaces implemented by or used by the first assembly and any of the components, the second assembly data-structure determining the components required to provide access to the second service and the relationships between the first assembly and the components, whereby the second assembly ~~definition~~ data-structure is configured to be loaded into the assembly manager, the assembly manager being operable to load and assemble the components at run-time according to the metadata information to provide access to the second service.

20. (Currently amended) The product of claim 17, further comprising instructions operable to cause the processor to:

modify one of the components;

create a new component to filter information that passes through an interface connected to the modified component; and

modify the first assembly ~~definition~~ data-structure to specify the new component and an interface connecting the new component to the modified component, whereby the modified component and new component so connected produce filtered information compatible with other components in the first assembly.

21. (Currently amended) The product of claim 20, wherein modifying the component alters the processing of information and renders the modified component and information incompatible with the other components in the first assembly.

22. (Currently amended) The product of claim 17, wherein the instructions operable to cause the processor to define the first assembly comprise instructions operable to cause the processor to identify client and server relationships between the components and interfaces.

23. (Currently amended) The product of claim 17, wherein the first assembly ~~definition~~ data structure is represented using Extensible Markup Language (XML).

24. (Original) The product of claim 17, wherein the components and interfaces comply with an object model architecture.

25. (Original) The product of claim 24, wherein the object-model is selected from a set of object-models including Component Object Model (COM), Bravo Interface Binder (BIB), and Common Object Request Broker Architecture (CORBA).

26. (Currently amended) A computer program product, tangibly stored on a computer-readable medium, for organizing components to provide access to a service by a component-based application, the product comprising instructions operable to cause a programmable processor to:

receive a request from the component-based application that identifies a service;

access an assembly ~~definition~~ data-structure associated with the service and having metadata information specifying a number of components used to perform the service and interfaces implemented by and used by the components, the assembly data-structure determining the components required to perform the service and the relationships between the components;

load each component identified in the assembly data-structure into an area for processing; and

connect an interface associated with each loaded component to other components according to the meta-data information in the assembly ~~definition~~ data-structure to form an assembly, whereby the application has access to an interface for communicating with the assembly.

27. (Currently amended) The product of claim 26, further comprising instructions operable to cause the processor to:

connect interfaces identified in the assembly ~~definition~~ data-structure to the loaded components; and

connect interfaces associated with components in the assembly ~~definition~~ data-structure but not identified in the assembly ~~definition~~ data-structure to the loaded components.

28. (Original) The product of claim 27, further comprising instructions operable to cause the processor to connect interfaces in the assembly to components in a previously loaded assembly.

29. (Currently amended) The product of claim 26, further comprising instructions operable to cause the processor to:

receive an indication that the access to the requested service is not longer required;

disconnect the interface from each component associated with the requested service; and

unload each disconnected component and the corresponding assembly ~~definition~~ data-structure while the component-based application remains loaded.

30. (Currently amended) A computer program product, tangibly stored on a computer-readable medium, for gaining access to a service, the product comprising instructions operable to cause a programmable processor to:

~~identifying~~ identify a service for processing data;

~~calling~~ call an assembly manager with a service request corresponding to the service, the assembly manager determining an assembly that is capable of performing the service; and

~~accessing~~ access [[an]] the assembly ~~capable of performing the service~~, the assembly including components and interfaces specified in an assembly ~~definition~~ data-structure and loaded by the assembly manager, the assembly data-structure determining the components required to perform the service and the relationships between the components, the assembly manager being operable to assemble the components at run-time according to information in the assembly data-structure to perform the service.

31. (Currently amended) The product of claim 30, wherein the service request comprises a name associated with the assembly ~~definition~~ data-structure.

32. (Original) The product of claim 30, wherein the service request comprises a role name associated with the service.